Hindawi
Journal of Sensors
Volume 2018, Article ID 2089626, 17 pages
https://doi.org/10.1155/2018/2089626

Research Article

A Study on a Secure USB Mechanism That Prevents the Exposure of Authentication Information for Smart Human Care Services

Kyungroul Lee,[1] Insu Oh,[2] Yeunsu Lee,[2] Hyeji Lee,[2] Kangbin Yim,[2] and Jungtaek Seo[2]

[1]*R&DB Center for Security and Safety Industries (SSI), Soonchunhyang University, Asan 31538, Republic of Korea*
[2]*Dept. of Information Security Engineering, Soonchunhyang University, Asan 31538, Republic of Korea*

Correspondence should be addressed to Jungtaek Seo; seojt@sch.ac.kr

Received 31 January 2018; Accepted 27 August 2018; Published 29 October 2018

Academic Editor: Mucheol Kim

The secure USB flash drive was developed to improve the security of the conventional USB flash drive, which is vulnerable to leakages of internally stored data caused by extortion, loss, etc. However, it has been continuously reported that the secure USB flash drive, which protects data through the adoption of a wide range of security technologies in wide-ranging ways, cannot assure data security because of implementation and environmental vulnerabilities, eavesdropping, unlock commands, and reverse engineering. As such, there is growing demand for a more powerful secure USB flash drive to solve these fundamental problems. Therefore, this paper presents a secure USB mechanism that prevents leakages of authentication data and does not compare authentication data for smart human care services, which have been a fundamental problem of existing flash drives. The proposed mechanism provides better security than the existing secure USB flash drive by satisfying the need for confidentiality, integrity, authentication, and access control and safely protecting data from impersonation, man-in-the-middle, replay, and eavesdropping attacks by malicious attackers. An assessment of its security using the formalized verification tool AVISPA has proved that it is safe. Therefore, it is considered that a safer, more secure USB flash drive can be manufactured using the mechanism proposed in this paper.

1. Introduction

Generally speaking, a mobile storage unit USB flash drive is inserted into a USB port of a given host for use and is usually referred to as a USB memory stick or USB disk. The concept was originally introduced by an Israeli IT company in 2000 [1] and has been advanced significantly since then. Indeed, USB technology has been further developed because such devices offer easy connection and disconnection, free data change and deletion, fast data transfer, and high portability [2, 3]. In fact, these features of the USB flash drive have made it the most widely used storage unit currently in use [3].

The USB flash drive is mostly used for data storage and backup, booting disks, and portable program storage [3], and it must be capable of storing many different types of data in order to provide such functions. The various types of stored data include sensitive data such as public certificate, confidential business data, and personal information [3]. Therefore, a USB flash drive with guaranteed security is essential since serious damage can occur if such information is extorted by a malicious attacker.

Despite positive purposes, USB flash drives have problems regarding sensitive data leakage due to no additional security functions. Such a problem is the root cause of security threats that expose sensitive data through extortion and loss, and the reason for the threat is the fact that the data in the flash drive are stored in a raw form, rather than being encrypted in an altered form to cope with external attacks. Security threats by extortion enable a malicious attacker to extort a victim's private data stored inside a USB flash drive by seizing and inserting them into his own computer. There is a more serious problem in that there can be secondary or

tertiary victims if the extorted data are items of personal information such as public certificate or confidential business information. The security threat by loss refers to cases which a flash drive is lost as a result of a victim's error. As with the security threat by extortion, a third party who obtains a lost drive can take the data stored inside the drive by inserting it into his or her computer.

The secure USB flash drive was developed to solve the problem of data exposure by the security threats described above. A secure USB flash drive is a USB flash drive that comprises a security function designed to safely protect data stored inside a USB and supplements the vulnerability of a conventional USB flash drive with data encryption/decryption, user authentication and identification, prevention of arbitrary data copying, and data erasure technology to protect the data at the time of loss [3, 4]. There are many studies on diverse methods of applying these technologies to protect data, which can be broadly divided into the software type and the hardware type. The software type refers to methods of protecting internal data using only software and includes the data encryption/decryption method and the access control method [5]. The data encryption/decryption method uses a mathematical tool to encode and decode the data to prevent the inference of plain text, in order to solve the problem of data being stored in plain text in a conventional USB flash drive. Encryption/decryption methods include the encryption/decryption of a whole disk, the encryption/decryption of an image file to be used as a drive, and the encryption/decryption of a container file for stored files. Unlike the data encryption/decryption method, the access control method uses user authentication to block access by unauthorized users and allow access only by authorized users to the disk itself, image, or file used as a disk. Access control methods include access control [2] through user authentication [5] and access control by device authentication. On the other hand, the hardware method safely protects data using a separate hardware module. The main methods of protecting data include the method which uses an encryption module and the method which uses a flash drive controller [5]. The method using an encryption module uses a dedicated hardware security module to make it difficult to access secret information using reverse engineering, which is the fundamental problem of the software method. It includes the method of attaching a dedicated chip for user authentication and data encryption/decryption chip and a method of controlling access to the drive using a biometric authentication module to improve the security of user authentication. The method using a flash drive controller adds a security function to the existing flash drive controller to secure safety, instead of changing the hardware design so as to attach an encryption module. It includes the method of providing user authentication by adding a function for setting the password in the controller and the method of partitioning the flash drive into a general area and a secured area and performing access control and data encryption/decryption on the basis of the authentication data stored in the secured area, which cannot be accessed from outside [6].

Despite the protection of internally stored data using a secure USB flash drive which applies the above methods, there are still security problems related to data leakage to the outside due to such vulnerabilities as the leakage of a transferred password and authentication bypass or the exposure of the encryption/decryption key [2]. Such security threats to a secure USB flash drive are mainly classified into implementation and environmental vulnerabilities, eavesdropping, unlock commands, and reverse engineering. Implementation vulnerability refers to the design vulnerability that occurs when the security function of the management program that supports a secure USB flash drive has not been fully implemented. One example of this is the problem that occurs with the exception handling function when a public certificate is saved in a secure USB flash drive [3] and an unauthorized user accesses the data using the function that initializes the password and the data [7]. Moreover, another threat is caused by the incorrect implementation of data erasure by limiting the maximum password input count to protect the data at the time of loss. This enables an attacker to arbitrarily change the password input count limit and extort the password via a brute force attack. The environmental vulnerability is not the vulnerability existing in the secure USB flash drive but that which occurs because of the environment that connects the drive and the host or the host platform environment. For example, there is a potential threat of an unauthorized user accessing the important data stored in a secure USB flash drive when the secure USB is recognized by an installed VMware and directly accessed [3]. Other examples include the threat to the security function of a management program that is not operating properly following booting in the safe mode, forced termination of a management program, and the time difference during booting [3] and the threat of authentication bypass as a result of an exposed password or password hint by eavesdropping the data transferred between the secure USB flash drive and the host, because the security function was not considered when the USB interface was designed [6]. The vulnerability caused by an unlock function occurs by providing separate security domain connection and password initialization commands regardless of user authentication. Examples include the threat of accessing an inaccessible security domain by resending an unlock command or an unlock command with the authentication data [6]. The vulnerability caused by reverse engineering is an authentication bypass or encryption/decryption key exposure by the analysis of the security function provided by a management program [6].

There have been cases of internal data being exposed due to the failure of a secure USB flash drive to safely protect the data due to the various vulnerabilities described above, implementation vulnerability cases of abusing the initialization function, environmental vulnerability cases of eavesdropping, and unlock command vulnerability cases of password initialization. The implementation vulnerability case of using the initialization function includes the case of data not being deleted even after the initialization of the management program provided by the company "S" and thus the data being recovered with a data recovery tool in order to access the internally saved data without authentication [7]. The environmental vulnerability case of eavesdropping includes eavesdropping of plain text data transferred in

communication between the secure USB flash drive of companies A, S, L, and I and the host to capture the password or password hint used in user authentication [7]. The unlock command vulnerability case using password initialization includes normal login to a banking site using the reset password changed by sending the password reset command and the reset password to the secure USB flash drive for public certificate of company C. Such cases show that there is a serious risk of important data stored in secure USB flash drives being exposed and such simple data exposure can lead to serious monetary damages.

As such, it is urgently necessary to establish measures for manufacturing more powerful secure USB flash drives due to the failure of secure USB flash drives fitted with security technology to protect data. Therefore, this paper proposes a safe security USB mechanism that does not expose software-based authentication data in order to improve the security of existing secure USB flash drives. The fundamental problem with existing secure USB flash drives is that it is possible to access the data by bypassing authentication via exploiting the exposure of authentication data or modified authentication data stored within the drive and the existence of a routine that allows comparing authentications in the management program of the host. The mechanism proposed in this paper does not compare the authentication data and does not store the authentication data inside the drive. As such, it provides stronger security than existing secure USB flash drives by preventing data exposure through the authentication bypass caused by the abovementioned problems, thus satisfying the confidentiality, integrity, authentication, and access control requirements and safely protecting data from impersonation, man-in-the-middle, and eavesdropping attacks.

This paper is organized as follows: Section 2 reviews the classification, security requirement, and security technologies of the USB flash drive, which is the background of this study, and investigates vulnerabilities that cause data leakage in order to check the problems of existing secure USB flash drives, Section 3 describes the proposed mechanism and evaluates its security, and Section 4 presents the conclusion and outlines future studies.

2. Related Studies

The original secure USB flash drive was designed to safely store important information and assure the security of internally stored data at the time of loss, by applying hardware or software-based data encryption/decryption, user authentication and identification, prevention of arbitrary data copying, and data erasure technologies. However, despite the application of such wide-ranging security technologies, vulnerabilities due to the improper implementation of the security functions or the environmental limitations of the current platform have been reported and such vulnerabilities have led to such problems as stored data being leaked or extorted to the outside by a malicious attacker. Therefore, this study reviews the overview, classification, security requirement, and security technologies of existing USB flash drives as the background knowledge and investigates vulnerabilities that cause data leakage in order to identify the problems of existing secure USB flash drives.

2.1. Overview of Secure USB Flash Drives. A secure USB flash drive is a USB flash drive that protects the important data stored inside it using hardware or software technology [8]. Unlike conventional USB flash drives, the secure USB flash drive provides such security functions as user authentication and separates the flash drive into the general domain and secured domain for that purpose [9]. The secured domain stores special data such as authentication data for user authentication and the encryption/decryption key for data encryption/decryption. The general domain in which the user data are stored is safely protected by security technology such as user authentication and access control based on the special data. The security function through user authentication, for example, safely protects internally stored data by comparing the user authentication data stored in the secured domain with the requested authentication data to block access to the general domain by users who provide incorrect authentication data and allows access only by users who provide the correct authentication data. Such a security function is provided by attaching an additional hardware module that performs the security function or by applying software technology only. The method of attaching a hardware module uses the special protocol defined by the manufacturer to perform the security function, as the USB protocol failed to reflect the security requirement when it was first designed [3, 4]. The method of using software only runs special software implemented for security in the host to provide the security function. The various methods of safely protecting the data stored in a secure USB flash drive can be classified in the ways outlined below.

2.2. Classification of Secure USB Flash Drive Type. As described above, secure USB flash drives are classified into the hardware and software types. The hardware type of method uses a hardware module that provides an additional security function, such as user authentication or data encryption/decryption, while the software type of method uses the special software provided by the manufacturer as an additional security function, such as access control [5]. Each type is further classified in detail, as shown in Table 1.

2.2.1. Classification of Hardware Type. The hardware type adds a dedicated hardware module as the security function to safely protect the data and is divided into the controller method, which adds a security function such as user authentication in the controller of the existing USB flash drive and the encryption module method, which uses a dedicated encryption module to provide the security function, such as data encryption/decryption [10]. Table 2 shows the detailed classification of the hardware type.

The flash drive controller method assures the security by adding a security function to the existing flash drive controller instead of changing the hardware design by attaching a new encryption module. It is further divided into the controller-internal password method and the partitioning method. The controller-internal password method provides

Table 1: Classification of secure USB flash drive types.

Classification		Method	Applied security technology
Hardware	Controller method	Controller internal password authentication	User authentication
		Partitioning	Access control
	Encryption module method	Biometric authentication	User authentication
		Data encryption module	Data encryption/decryption
Software	Disk utilization method	Whole disk	Data encryption/decryption
		Image file	Data encryption/decryption
		Container	Data encryption/decryption
		Reserved domain	Access control
	Access control method	User authentication	Access control
		Device authentication	Access control

Table 2: Classification of hardware type.

Classification	Method	Description
Controller method	Controller internal password authentication	User authentication using password
	Partitioning	Access control using partitioning
Encryption module method	Biometric authentication	User authentication using biometric data such as fingerprint
	Data encryption module	Data encryption/decryption using a dedicated encryption chip

Table 3: Classification of software type.

Classification	Method	Description
Disk utilization method	Whole disk	Data protection by encrypting/decrypting whole disk
	Image file	Data protection by encrypting/decrypting the image file to be used as a disk
	Container	Data protection using container file
	Reserved domain	Use of reserved area as the secured domain
Access control method	User authentication	Access control through user authentication
	Device authentication	Access control through device authentication

user authentication by adding a function to set the password in the controller and protects the data by only allowing authenticated users to access the data. With the user authentication technology, the users register the password and are allowed to access internally stored data only when the password input during authentication matches the registered password [5]. The partitioning method divides the secure USB flash drive into partitions to add domains that provide the security function and control access through them. The flash drive is partitioned into the secured domain to store the user authentication data and the general domain to store the user data; access to the general domain is allowed only when the authentication data input during the authentication process matches the stored authentication data [6].

The encryption module method uses a dedicated hardware security module to make it difficult to access secret data through reverse engineering, which is the fundamental vulnerability of the software method. It is further divided into the biometric authentication method and the data encryption module method. The biometric authentication method attaches a biometric authentication module to the secure USB flash drive to protect the data by encrypting/decrypting the flash drive through user authentication [10]. In other words, it assures the security of data stored in the flash drive by preventing users who fail the biometric authentication from using the flash drive and allowing only users who pass the biometric authentication to use the flash drive. The encryption module method provides data confidentiality by attaching a dedicated encryption module which performs user authentication and data encryption/decryption. It prevents the leakage of original data by authenticating users with the encryption module and by encrypting/decrypting the data based on the encryption/decryption key generated inside the module for normal users and does not encrypt or decrypt the data otherwise. This method provides additional security by permanently deleting internally stored data when authentication fails a specific number of times or when the flash drive is disassembled by force [10].

2.2.2. Classification of Software Type. The software type safely protects the internally stored data using only software. It is divided into the disk utilization method using encrypting/decrypting [5] and the access control method through user authentication and device authentication [2]. Table 3 shows the detailed classification.

Table 4: Classification of security technology of secure USB flash drives.

User authentication technology	Data security technology		
User authentication and identification	Data encryption/decryption	Data erasure for protection after loss	Prevention of arbitrary data copying
Password	On-the-fly encryption	Tampering	Access control
Biometric authentication	Selective file encryption	Input count limitation	Management system

The disk utilization method is classified not according to the data protection type but rather according to the subject that is used as the disk. It is further divided into the method of using the whole disk, that of using an image file as a disk, that of using a container file for the data and security function, and that of using the reserved area of the system as a disk [10]. The method of using the whole disk protects the data by encrypting and decrypting the whole disk using the software provided by the manufacturer. It prevents the leakage of original data stored throughout the disk. More specifically, this method decrypts the whole disk for use as a conventional USB flash drive when an authorized user wants to use the disk and decrypts the whole disk to safely protect the internally stored data when the user terminates use of the disk [5]. The method of using an image file does not encrypt and decrypt the whole disk but instead generates an image to be used as a disk first and then encrypts and decrypts the generated image to protect the data. It not only prevents leaks of data stored in the drive but also quickly encrypts and decrypts the data since it does not process the whole disk [2, 10]. Although the detailed process of this method is the same as the method of using the whole disk, the difference here is that the subject is not a whole disk but an image to be used as a disk. The concept of the method of using a container file is similar to that of the hardware-based partitioning method, except that it uses the container file which partitions the domains with software instead of hardware partitioning. Thus, it safely protects data by using the container file to store the data needed for user authentication and encrypting and decrypting both the container file and data [5]. The method of using a reserved area stores secret data such as authentication data in a reserved space inside a USB flash drive and controls access to the stored data based on it. It prevents data access through an authentication bypass by concealing the authentication data or transformed authentication data in the reserved space [2, 10].

The access control method does not encrypt/decrypt the data stored inside a flash drive but allows access to the flash drive or an image to be used as a disk only by authorized users through user authentication and device authentication. Access control through user authentication controls access to internal data by authenticating users with the software provided by the manufacturer. It compares the registered authentication data and the requested authentication data and allows access only when they match and blocks access when they do not match, in order to assure safety [5]. Device authentication allows or blocks the use of a device based on unique device information such as the serial number. It prevents the leakage of confidential data by only allowing the use of an accepted device in the accepted space and by blocking the use of an unaccepted device in an accepted space or an accepted device in an unaccepted space based on the in/out policy. In addition, it can be combined with the access privilege of the user to prevent leakages of internal data by allowing only authorized users to use it, and even then only when the accepted device is connected in an accepted space, and by blocking its use if an accepted user does not have the privilege.

2.3. Security Technology of Secure USB Flash Drives. The secure USB flash drive is configured in the diverse ways described above, and each method safely protects the internal data by applying a wide range of security technologies. Such security technologies must satisfy the requirements of user authentication and identification, data encryption/decryption, and data erasure so as to protect the drive at the time of loss and also prevent arbitrary data copying [11]. Table 4 shows the leading security technologies designed for that purpose.

2.3.1. User Authentication Technology. User authentication technology identifies authorized users and allows only authenticated users to use services and is also known as electronic authentication technology in the computing environment. The leading electronic authentication technologies include knowledge authentication based on the user's memory, such as the password, possession authentication based on the medium possessed by the user, and biometric authentication based on the user's unique physical data [12]. User identification is checked using a wide range of factors. The most widely used electronic authentication technology is the password-based knowledge authentication technology, which performs registration and authentication based on the user-remembered password. Since this technology is dependent upon the user's memory, it does not require a separate medium and has the benefits of low cost and high user friendliness. Such benefits make it the preferred user authentication technology for secure USB flash drives, and most commercial secure USB flash drives authenticate users with the password. However, there is a possibility of the password being exposed since the passwords registered by authorized users are generally stored inside the flash drive; but a more serious problem is the fact that the routine procedure by which the registered password is compared with the input password exists on the host side, making it vulnerable to exposure of the password and bypassing of authentication through reverse engineering [6]. To solve the vulnerability of the password method, products that run the password-comparing routine in a separate module attached inside the flash drive have been developed. However, these products

also store the authentication data inside the flash drive and so have the fundamental problem of not being able to control access inside the flash drive and thus cannot prevent authentication bypass by password change and exposure.

To solve the fundamental problem of authentication data exposure, the method of attaching a biometric authentication module in the flash drive in order to activate the flash drive through biometric authentication has been developed. Fingerprint recognition technology is the most widely used biometric authentication method developed to supplement the problem with the password method. Users register their fingerprint through the fingerprint recognition module attached to the flash drive, and access to the flash drive is only allowed to the user whose inputted fingerprint matches the registered fingerprint. Despite this technological advance, the problem of extorting the fingerprint remains because authentication bypass through reverse engineering, like the password method, is not impossible although it is not easy [10].

2.3.2. Data Security Technology. The data stored inside a secure USB flash drive can be leaked if the flash drive is lost by extortion by a malicious attacker or by a user error if the flash drive is not equipped with a data security technology. For that reason, the data security technologies are applied to prevent data leakage to outside and uncontrolled data access. The various technologies include a data encryption/decryption technology designed to prevent leakages of original data, a technology for preventing arbitrary data copying to control access to the data, and a data erasure technology for protecting a flash drive from data leakage after a loss.

Data encryption/decryption technology generates a cryptogram from the plain text data using an encryption algorithm to protect the original data. The technology is divided into full disk encryption (FDE) and on-the-fly encryption (OTFE). The method of encrypting the whole disk encrypts and decrypts all data stored on the disk according to the user authentication result. All data are decrypted in order to use the flash disk as a conventional disk if the user authentication is successful, and the whole disk is encrypted again to prevent the leakage of original data after use [5]. However, as there is a possibility of data leakage as the data are saved in their original form after user authentication, the OTFE was developed to solve the problem by preventing data leakage even while the disk is in use. This method prevents the data leakage inherent to FDE since it only decrypts data to be used by the user, instead of the whole disk, and leaves unused data in the encrypted form even after successful user authentication [5].

Because of its innate characteristics, a USB flash drive is highly portable and there is a high risk of loss due to user error, and an attacker who possesses a lost drive can separate the memory storing the data and obtain the internally stored data using a specially produced tool. For that reason, an antitampering technology designed to detect malicious hardware manipulation of flash drives was developed to protect data in the event of loss of a flash drive. This technology permanently deletes the internally stored data to prevent data leakage if a malicious hardware disassembly or manipulation is detected. However, a user authentication bypass can occur through a brute force attack that enters all possible passwords to obtain the user passwords, instead of hardware manipulation. For that reason, a technology that limits the input count was developed to prevent brute force attacks. This technology permanently deletes the internally stored data to prevent data leakage if the specific input count is exceeded.

User authentication technology has a serious vulnerability in that authentication bypass can be achieved by extorting the stored authentication data or updating them with altered authentication data, since the authentication data are stored inside the flash drive. Since the root cause of this vulnerability is the accessibility to the data stored inside the flash drive, technologies for preventing the arbitrary copying of data have been developed and these can be divided into a technology for controlling access to data and a management system that allows or blocks use of the device. This access control technology validates the subjects accessing the data through user authentication and device authentication. If authentication fails, connection to the memory storing the data is blocked via a hardware method or the data are concealed by a software method of disabling data access. If authentication succeeds, the above methods are inactivated to allow data access. However, there is a problem in that original data can be acquired by bypassing authentication as the data are not encrypted for storage if only access control is provided [5]. On the other hand, the technology consisting of a management system that controls access to a device (instead of access to data) prevents the arbitrary copying of data by allowing or blocking the use of the device according to user authentication, user privilege, and the space where the device is used based on the established policy. In detail, the technology registers the user authentication data and privilege and the devices to be used and then allows the use of the device only to authorized users who pass the authentication, but only when the registered device matches. The use of the device is blocked to prevent the leakage of internal data if any of the registered data do not match. In addition, leakages of internal data can be prevented by allowing the use of devices only in registered safe areas, i.e., the internal network, and by blocking use from outside, in order to improve security [13].

2.4. Vulnerability of Secure USB Flash Drives. Even after applying the abovementioned security technologies to the secure USB flash drive, vulnerabilities related to external leakage of the data safely stored inside the flash drive have been detected. The main vulnerabilities include the implementation vulnerability, environmental vulnerability, unlock command, and reverse engineering [3]. Table 5 shows the vulnerabilities detected so far and the security technologies that have been disabled by the identified vulnerabilities.

The implementation vulnerability is a design vulnerability that occurs when the security function of the management program designed to support the secure USB flash drive is not properly implemented. Examples include public certificate bypass, password initialization, and input count manipulation [3, 7]. Public certificate bypass accesses the data using exception handling when the public certificate is stored in a secure USB flash drive. Specifically, since the

Table 5: Classification of vulnerabilities of secure USB flash drives.

Vulnerability	Classification	Description	Security technology
Implementation vulnerability	Exception of public certificate	Exception handling when the public certificate is stored to allow data access	(i) Prevention of arbitrary data copying
	Password initialization	Data access through recovery tool after password/data initialization	(ii) Data protection after loss
	Input count manipulation	Manipulation of password input count to infer password through brute force attack	(iii) Data protection after loss
Environmental vulnerability	VMware	Disabling of management program using VMware	(iv) User authentication and identification
	Direct memory access	Direct access to flash drive to read and write data	(v) Prevention of arbitrary data copying
	Safe mode	Booting in safe mode to disable the management program	(vi) User authentication and identification
	Forced termination	Termination and disabling of management program by force	(vii) Prevention of arbitrary data copying
	Booting time difference	Data access during the time the management program is not run in the booting process	(viii) Prevention of arbitrary data copying
	Eavesdropping	Analysis of data transferred between the host and the flash drive to obtain the password or password hint	(ix) Data encryption/decryption
Unlock commands	Secured domain access command	Sending of unlock command to access the secured domain	(x) User authentication and identification
	Command containing the authentication data	Sending of unlock command containing the authentication data to access the secured domain	(xi) User authentication and identification
Reverse engineering	Authentication bypass	Analysis of authentication process of management program to bypass authentication	(xii) User authentication and identification
	Exposure of encryption/ decryption key	Analysis of data encryption/decryption function of management program to obtain the encryption/decryption key	(xiii) User authentication and identification (xiv) Data encryption/decryption

public certificate stored inside a drive can be accessed without the approval of a secure USB flash drive, the internal data can be changed to .pfx, which is an extension of the public certificate, using this function, and then arbitrarily copied [3]. Password initialization allows an unauthorized attacker to access the data using the password and data initialization function. In detail, the attack uses the fact that the password initialization function, which deletes the internally stored data when the secure USB flash drive is reset to the initial condition, does not permanently delete the data. The data can be obtained using a recovery tool that restores the data [7]. Input count manipulation concerns the incorrect implementation of data erasure by limiting the maximum password input count to protect the data at the time of loss. It enables an attacker to arbitrarily change the password input count limit and extort the password through a brute force attack. In detail, it manipulates the password input count limit in the management program to infinity through reverse engineering and then tries all possible passwords to obtain the registered password.

Environmental vulnerabilities are not present in a secure USB flash drive itself but occur because of the environment between the drive and the host or the host environment. Examples include vulnerabilities of VMware, direct memory access, safe mode, forced booting, booting time difference, and eavesdropping. The vulnerability of VMware allows data access by recognizing the secure USB flash drive in the VMware, so that the management program providing the security function does not run. It can cause serious damage as it does not need to disable or bypass the security program and does not leave a usage record [3]. The vulnerability of direct memory access is provided by the operating system and allows an attacker to access the data by accessing the drive directly through implementation of or by using a tool instead of accessing data via the inside of the flash drive [3]. Other vulnerabilities that enable attackers to access internally stored data include booting in the safe mode or terminating the management program by force, so that the management program that provides the security function does not run, using the time before the security program is run during booting [3]. Lastly, the eavesdropping attack exploits the vulnerability caused by failure to consider the security function in the design of a USB interface so as to bypass authentication by obtaining the password and password hint transferred between the flash drive and the host, extorts the data by obtaining the transferred encryption/decryption key, and

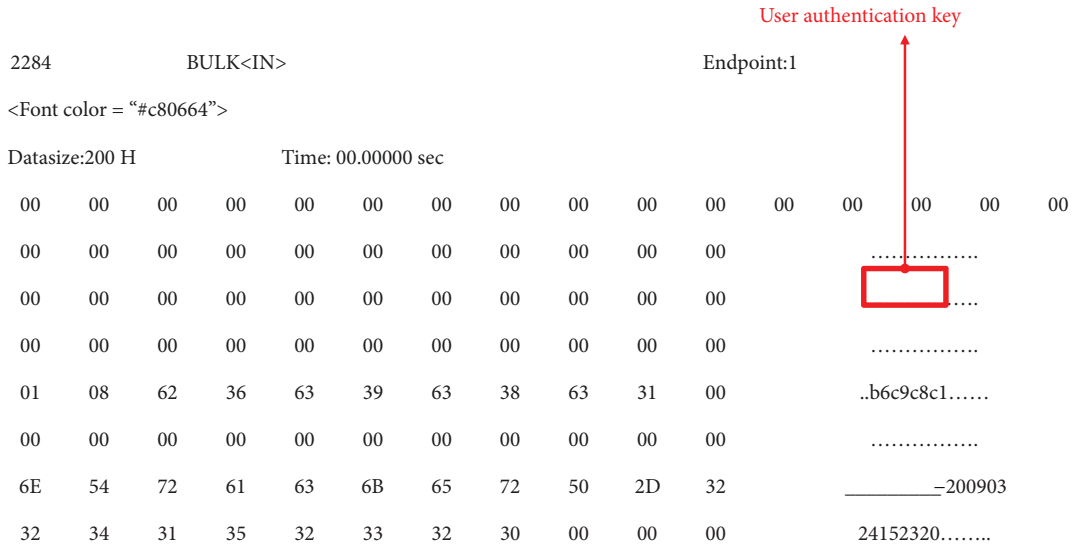

Figure 1: Example of authentication data exposure.

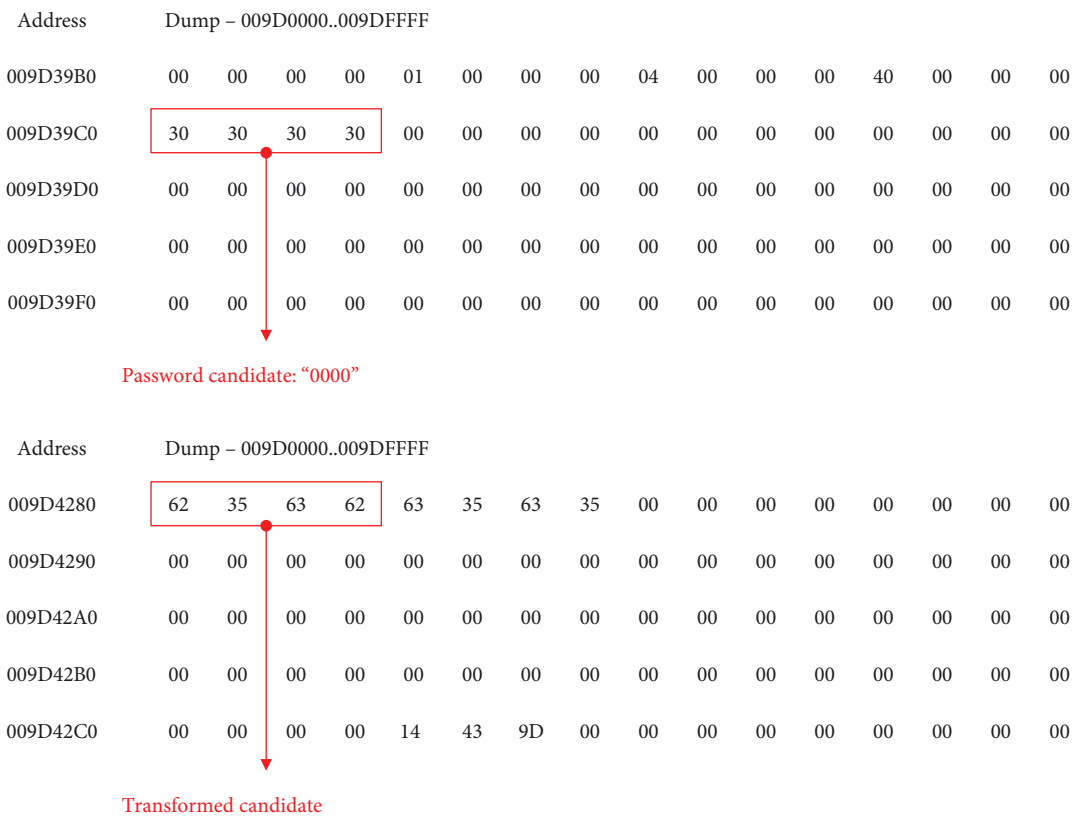

Figure 2: Example of use of exposed authentication data in management program.

then decrypts the encrypted data [6]. There have been actual cases of data being exposed in this way during communications, as shown in Figures 1 and 2.

The secure USB flash drive which separates the general domain and the secured domain in the hardware method has a special unlock command that connects the secured domain when user authentication is successful. In such a case, an attacker can analyze the unlock command and directly send a command enabling connection to the secured domain and extortion of internally stored data [6]. Such an unlock command can be configured by only the command code or additionally demanding the authentication code.

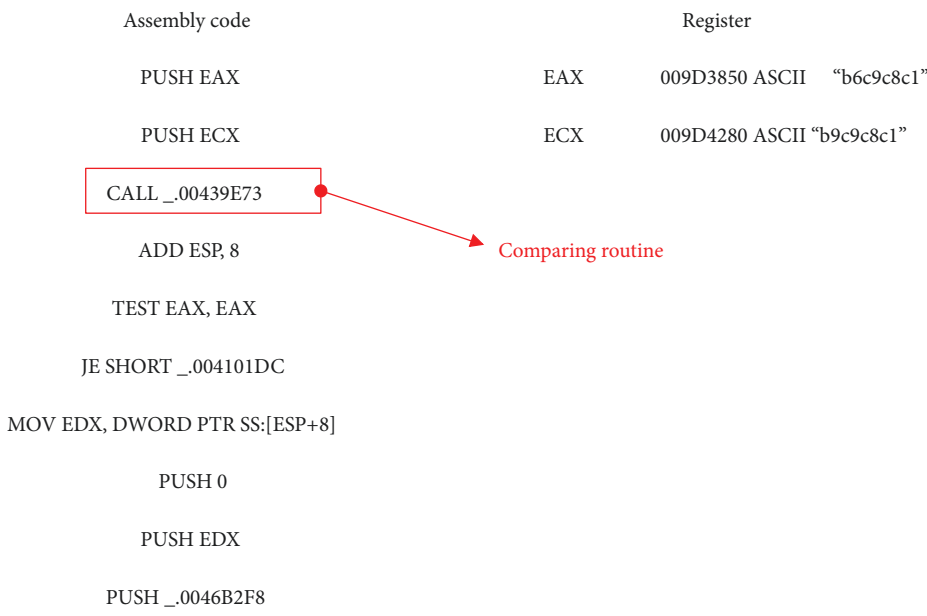

Figure 3: Example of comparison routine using exposed authentication data.

Although access to a secured domain is not possible if the correct authentication data are unknown, an additional command that transfers the authentication data is provided. As such, there have been cases in which the authentication data and the secured domain-connecting command were sent together to extort the internally stored data [8].

The vulnerability of reverse engineering allows an attacker to bypass user authentication and extort the exposed encryption/decryption key by analyzing the security function of the management program using reverse engineering. There have also been actual cases in which the routine that compares the authentication data and authentication bypassed by manipulating the routine in order to approve user authentication by force has been analyzed, as shown in Figures 3 and 4, or in which the encryption/decryption process has been analyzed, the exposed encryption/decryption key acquired, and the encrypted data decrypted [6].

Problems related to the failure of commercial secure USB products to safely protect internally stored data have been detected due to the various vulnerabilities described above. Therefore, this paper presents a mechanism that is designed to supplement the vulnerabilities of existing secure USB flash drives without exposing the authentication data in order to solve these problems.

3. Proposed Mechanism

Existing secure USB flash drives have problems in that the internally stored data can be extorted by exposing the password and bypassing user authentication due to the implementation vulnerabilities of public certificate exemption and password initialization; environmental vulnerabilities such as VMware vulnerability, direct memory access, and eavesdropping; and vulnerabilities using the unlock command. A more serious problem than any of the above vulnerabilities is that the current platform is based on the von Neumann architecture and loads the program code and the data needed for program operation in the memory. This allows program analysis through reverse engineering, which can lead to the exposure of key data. In other words, an attacker can analyze the internal operating process by reverse engineering the management program of a secure USB flash drive in order to extort the password and bypass authentication to access the data stored inside the drive. As an example, the management program run in a host performs the role of intermediary for delivering the authentication data between the secure flash drive and the user and storing the authentication data. Therefore, the key data must be allocated in the resource in which the management program is run. For that reason, the vulnerability which allows an attacker to access the resources containing the key data and extort the password or password-related authentication data and the architectural vulnerability which allows an attacker to bypass authentication, as the routine of comparing authentications exists in the management program of the host, have been identified. Since vulnerabilities with the flash drive due to the abovementioned fundamental reasons have been continuously discovered, studies are needed to remedy these problems.

Therefore, this paper proposes a safe software-based secure USB mechanism that disables authentication bypass while not exposing the authentication data in order to solve the aforementioned problems. The proposed mechanism uses a mathematical cryptologic tool to encrypt and decrypt the data and provides a function for authenticating users without exposing the authentication data, as well as preventing arbitrary copying of data, and erasing the data for protection after the loss of a flash drive. Table 6 shows the terms used in the proposed mechanism.

The proposed protocol consists of the registration process and the authentication process and provides disk management, user authentication, and data encryption/

1. Data encryption/decryption key is exposed

Address	Mem Dump															
0012E978	4D	55	42	5A	46	50	51	4E	57	43	57	47	53	57	5A	47
0012E988	49	59	55	45	56	48	41	52	56	52	48	4C	48	45	48	59
0012E998	46	41	57	46	56	51	43	44	43	45	50	45	4a	50	59	46
0012E9A8	59	44	55	53	42	53	41	46	45	5f	4b	45	59	49	43	45

ASCII: MUBZFPQNWCWGSWZGIYUEVHARVRHLHEHY…

2. Attacker inputs a dummy password

Address	Mem Dump															
0012D328	24	55	1b	0D	FA	93	7E	44	88	39	64	1F	A4	02	92	37
0012D338	C6	EA	D2	B9	C2	87	75	A3	04	23	D8	D2	71	6B	CF	FD
0012D348	A3	B3	D5	9A	24	32	38	6B	DD	AC	2D	65	4D	D9	53	F7
0012D358	7B	6A	67	1A	B2	F1	DD	A0	3F	0A	78	A6	C4	A5	C0	D3

3. Attacker replaces the key with the exposed one

Address	Mem	Dump														
0012D328	24	55	1B	0D	EA	93	7E	44	88	39	64	1F	24	02	92	37
0012D338	C6	EA	D2	B9	CB	87	75	A3	04	23	D8	31	71	6B	CF	FD
0012D348	A3	B3	D5	9A	24	32	38	6B	DD	AC	2D	65	4D	D9	53	F7
0012D358	7B	6A	67	1A	B2	F1	DD	A0	3A	0A	78	A6	C4	A5	C0	D3

Address	Mem	Dump														
0012E978	4D	55	42	5A	46	50	51	4E	57	43	57	47	53	57	5A	47
0012E988	49	59	55	45	56	48	41	52	56	52	48	4C	48	45	48	59
0012E998	46	41	57	46	56	51	43	44	43	45	4D	45	4a	50	59	46
0012E9A8	59	44	55	53	42	53	41	46	45	5F	4b	45	59	49	43	45

4. Authentication is by passed

Figure 4: Example of encryption/decryption key exposure.

decryption functions to assure data security. Such functions guarantee the security of the proposed mechanism by satisfying the requirement for confidentiality, integrity, authentication, and access control and by safely protecting the data from impersonation, man-in-the-middle, and eavesdropping attacks.

3.1. Registration Process. The registration process of the proposed mechanism consists of the following two steps: The first step is to register the users and encrypt the DIF, which is generated with the user-input authentication data and disk data. Specifically, the management program is run for registration and authentication in the host and the management program generates a DIF based on the authentication data received from a user and the data needed to generate the DIF. The user ID, which is needed for user authentication in the second step, is inserted in a reserved space of the file system header in the generated image file, and the generated image file is encrypted using the hash value of the user PW as the key, thus completing the first step of user registration. The second step authenticates a registered user by decrypting the encrypted DIF with the user-input authentication data and compares it with the registered authentication data to authenticate the user. The key to the user authentication process is to compare the ID inserted in the file system header during the registration process and the ID received from the authentication process to validate the user. The process decrypts the file system header encrypted with the hash value of the PW received in the authentication process and compares the received ID and the decrypted ID. The user is validated and registration is completed when the IDs match. Figure 5 shows the proposed registration process.

Table 6: Terms.

Term		Description
Status information	U	User
	ID	ID (identifier)
	ID'	ID rereceived by a user for disk decryption
	PW	Password
	PW'	PW rereceived by a user for disk decryption
	DIF	Disk image file
	ENDIF	Encrypted disk image file
	DEDIF	Decrypted DIF
Operating status	F	Format
	RUR	Request user registration
	RAI	Request authentication information
	TAI	Transfer authentication information
	IU	Identify user
	RGDIF	Request generation of DIF
	TDI	Transfer disk information
	GDIF	Generate DIF
	IID	Insert ID in an empty space in the file system header
	AU	Authenticate user
	FR	Finish registration
	MD	Mount disk
	SDIF	Select DIF
Encryption/ decryption	$h()$	Hash operation
	H	h(PW), result of hash operation based on PW
	H'	h(PW'), result of hash operation based on PW'
	$C = E_K(P)$	Result C of encryption of the plain text P based on the key K
	$P = D_K(C)$	Result P of decryption of the cyphered data C based on the key K
	EDIF	Encrypt DIF
	RDDIF	Request decryption of DIF
	RDFSH	Request decryption of file system header
	DFSH	Decrypt file system header
	DDIF	Decrypt DIF
Entity	User	User authentication
	Management program	Management program
	USB flash drive	USB flash drive

Step 1. A user requests registration with the management program when using the USB flash drive and then the management program confirms it (RUR).

Step 2. The management program requests the authentication data such as ID and PW, from the user requesting registration (RAI).

Step 3. The user inputs and sends the memorized ID and PW requested by the management program (TAI).

Step 4. The management program checks if the ID received from the user is in its database and approves the user's registration if it does not exist in the database (IU). Step 2 is repeated if the received ID exists in the database.

Step 5. If the user registration is approved, the management program requests generation of an image file to be used as a disk (RGDIF) and demands the data, such as the disk name and size of the DIF to be generated.

Step 6. The user inputs the DIF and the related data requested by the management program and sends them to the management program (TDI).

Step 7. The management program generates a file to be used as the disk inside the USB flash drive based on the disk image file and the related data received from the user (GDIF).

Step 8. The disk is formatted to add the data for disk recognition after the file to be used as a disk has been generated in the USB flash drive (F).

Step 9. The proposed mechanism uses the ID as the data for authentication in order to assure that the authentication data are not exposed and the ID received from the user (ID') is inserted into the reserved space of the header, such as the boot record of the file system (IID). The inserted ID is used to verify the user afterward. In detail, the inserted ID is encrypted with the hash value of the PW in the registration process and is decrypted with the hash value of the PW' received from the user in the later authentication process in Step 15. The user is authenticated in Step 15 by comparing the user-input ID' with the currently inserted ID.

Step 10. The management program performs the hash operation of the PW received from the user to generate the key for encryption of the image file to be used as a disk ($H = h(PW)$). The hash operation is necessary since using the PW directly as the key for encryption/decryption has the problem of PW leakage, because the time the PW is exposed outside increases during the encryption/decryption process. Performing the one-way hash operation prevents inference of the PW even when a third party obtains the hash value.

Step 11. The management program begins encryption of the DIF in the flash drive using the hash value (H) calculated in Step 10 ($ENDIF = E_K(DIF)$).

Step 12. The management program continues Step 11 with the whole DIF (ENDIF).

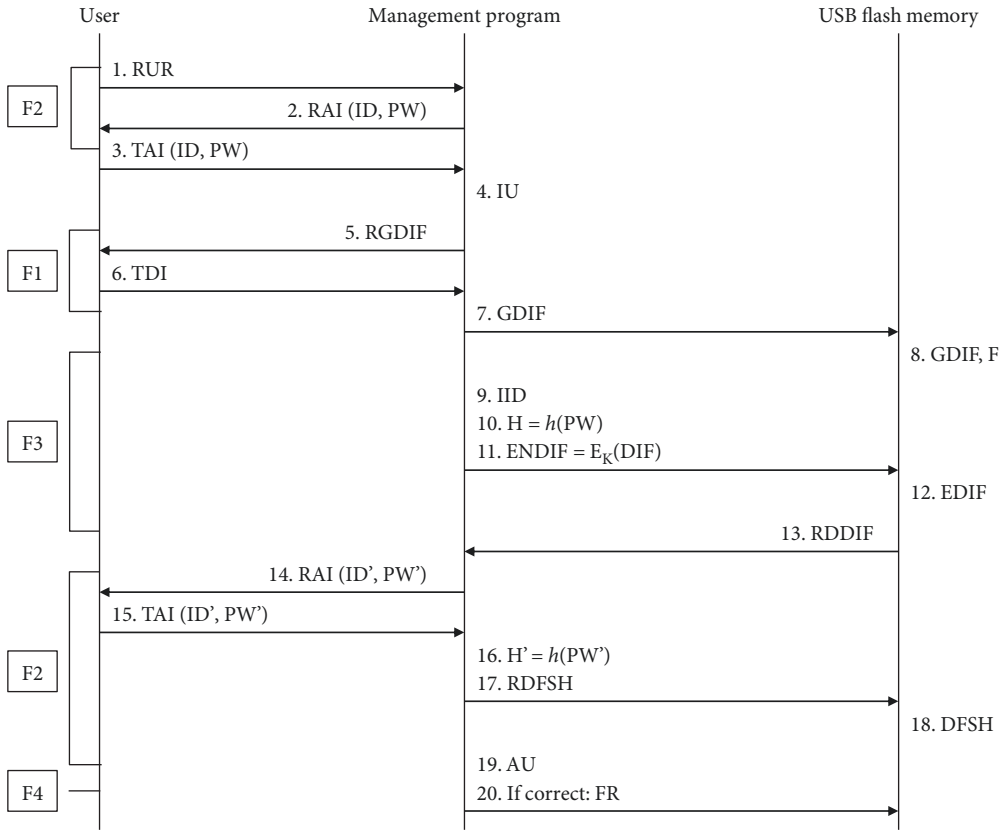

Figure 5: Proposed mechanism registration process.

Step 13. The first registration process is completed after Step 12. However, the proposed mechanism performs the user authentication process to validate the registration process. The management program requests decryption of the disk to the flash drive for user authentication (RDDIF).

Step 14. The management program requests the authentication data ID' and PW' from the user since the user ID and PW are needed to decrypt the disk (RAI).

Step 15. The user sends the ID' and PW' that he/she has registered (TAI).

Step 16. In the registration process, the user is authenticated by comparing the ID' input by the user in Step 15 with the ID inserted into the file system header in Step 9. To extract the inserted ID, the ID inserted in the file system header of the encrypted disk file must be decrypted and the hash operation is performed using the received PW' to generate the decryption key (H' = h(PW'))

Step 17. The management program requests a decryption of the file system header using the hash value (H') calculated in Step 16 (RDFSH).

Step 18. The decryption of the encrypted file system header of the disk image file is performed using the H' calculated with the user-input PW' as the key (DFSH).

Step 19. After the decryption of the encrypted file system header, the ID in the decrypted header and the ID' received from the user are compared for user authentication. The user is authenticated as a normal user if they match (AU) and the registration process is completed (FR). On the other hand, it is judged that an invalid password has been input if they do not match and the registration is canceled. The registration can be canceled by failed user authentication not only because of a malicious attacker attempting authentication bypass but also because of an error by a normal user. In that case, canceling registration with just an error will be very inefficient since the user will have to repeat the whole process starting from Step 1. To supplement this inefficiency, the user can return to Step 14 and input the authentication data again instead of repeating the whole process from Step 1. However, the process will be vulnerable to a brute force attack if infinite reentries of the authentication data are allowed. The input count can be limited to improve security, and the proposed mechanism limits the input count to 5 since the general input count is 5. If the input count exceeds 5, the user is judged to be a malicious user, and thus, the registration is canceled and all data are deleted.

The objects that participate in the registration process of the proposed mechanism are the user, the management program, and the USB flash drive. The data possessed by an object are deduced after the registration process is completed as the object information exposed to a third party can be used for malicious purposes. Although a user possesses the ID and

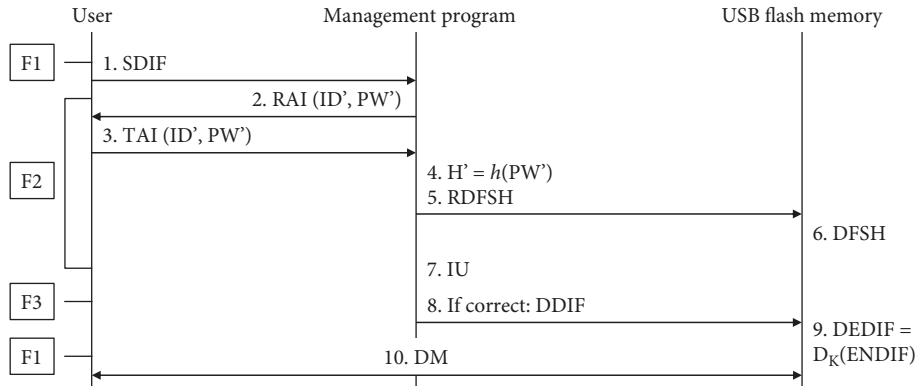

Figure 6: Proposed mechanism authentication process.

PW, which are the authentication data needed for registration, these data can be remembered only by the user and thus cannot be misused by attackers. Since the management program deletes the authentication data and the encryption/decryption data used in the registration process, there are no possessed data that can be misused by an attacker. Although the USB flash drive possesses the DIF encrypted with the hash value of the PW input by the user, the correct PW is remembered only by the user and the hash value of the correct PW cannot be inferred reversely because of the one-way characteristics of hash operation. Therefore, an attacker cannot abuse it. The applications using cryptology are all vulnerable to a brute force attack, but this problem can be partially resolved by limiting the input count and increasing the cryptologic strength.

When a user wants to use the disk after completing the above registration process, the following authentication process is performed to decrypt the encrypted DIF based on user authentication.

3.2. Authentication Process. The authentication process of the proposed mechanism allows only authorized users to use the encrypted disk. In more detail, user authentication is the process of authenticating the user by comparing the ID, which is the authentication data of the user requesting authentication, with the ID inserted into the file system header in the registration process. The file system header encrypted during the registration process is decrypted using the hash value of the PW received during the authentication process, and the user is authenticated when the data match. Then the whole disk file is decrypted and can be recognized as a general disk. Figure 6 shows the proposed authentication process.

Step 1. The user selects an EDIF to be decrypted from among the encrypted image files in the USB flash drive (SDIF).

Step 2. The management program requests the authentication data ID' and PW' to the user in order to decrypt the EDIF (RAI).

Step 3. The user sends the requested authentication data ID' and PW' to the management program (TAI).

Step 4. The management program performs the hash operation of the PW' input by the user in Step 3 to decrypt the encrypted file system header (H' = h(PW')).

Step 5. The management program requests the decryption of the file system header using the hash value (H') calculated in Step 4 (RDFSH).

Step 6. The decryption of the encrypted file system header of the disk image file is performed using the H' calculated with the user-input PW' as the key (DFSH).

Step 7. After the file system header has been decrypted, the ID inserted into the file system header in Step 7 is compared with the ID' input by the user in Step 3 to authenticate the user (IU). The user is authenticated and the whole DIF is decrypted (DDIF) if the comparison matches. Otherwise, it is judged that the IU has failed and the decryption of the DIF is canceled.

Step 8. After the decryption of the DIF, the file is mounted as a disk and can be used as a conventional disk after completing the mounting.

The above steps complete the whole authentication process. When the user no longer uses the disk, the decrypted flash drive is encrypted again so as to be protected safely. However, reusing the hash value generated in Step 4 can pose vulnerability since the key can be extorted by an attack, like memory scan, because the hash value of the PW, which is the encryption/decryption key, is continuously loaded in the memory while the disk is being used. Therefore, the encryption/decryption key allocated to the memory is deleted once the above authentication process has been completed. If the user does not use the disk afterward, the authentication data are requested again and used to encrypt the decrypted flash drive to protect the data when the drive is lost.

As with the registration process, the objects that participate in the authentication process of the proposed mechanism are the user, the management program, and the USB flash drive. The data possessed by an object are deduced since the object information exposed to a third party can be used

Table 7: Functions of the proposed mechanism and related steps.

Function	Detailed function	Related steps
F1 (disk management)	(i) File generation (ii) File opening (iii) File closing (iv) File deletion	(i) Registration process: Steps 5, 6, 7, and 8 (ii) Authentication process: Steps 1 and 8
F2 (user authentication)	(i) ID/PW (ii) Possible application of additional device authentication to improve security	(i) Registration process: Steps 1, 2, 3, 4, 14, 15, 16, 17, 18, and 19 (ii) Authentication process: Steps 2, 3, 4, 5, 6, and 7
F3 (data protection)	(i) Data encryption/decryption (ii) Saving of additional random number and use of hash chain to improve security	(i) Registration process: Steps 9, 10, 11, 12, 13, and 19 (ii) Authentication process: Step 7

for malicious purposes. Although a user possesses the ID and PW, which are the data needed for authentication, these data can be remembered only by the user and thus cannot be misused by attackers. Since the management program deletes the ID and password input by the user for decryption of the disk and comparison of the authentication data and the hash value of the PW, rather than storing them inside the management program after the completion of the authentication procedure, there are no data to be possessed, and thus, none can be used by an attacker. The USB flash drive does not save the user's PW and the encryption/decryption key during and after authentication; thus, an attacker cannot use the information.

The proposed mechanism provides disk management, user authentication, and data protection functions that use the registration process and the authentication process. It satisfies the requirements for data encryption/decryption, user authentication and identification, the prevention of arbitrary data copying, and the erasure of data for protection after loss of a drive, which are the four essential functions required to secure a USB as defined by the National Intelligence Service. Table 7 shows the steps related to these functions.

The data encryption/decryption requirement is satisfied with data protection function F3. To overcome the vulnerability of data being easily extorted because the original data are saved, as is the case in the existing USB flash drive, the image file to be used as a disk is generated and the generated DIF is encrypted and decrypted using the hash value of the user-input PW to prevent data leakage to the outside. The user authentication and identification requirement is satisfied with user authentication function F2. To overcome the vulnerability of data being extorted by authentication data exposure and authentication bypass of existing USB flash drives, it applies a measure to prevent the exposure of authentication data to prevent data leakage by authentication data exposure and authentication bypass. The prevention of arbitrary data copying requirement is satisfied by disk management function F1. After the image to be used as a disk is generated, it is mounted for use like a general disk only once the user has been authenticated. A user who fails the authentication procedure cannot use the disk since the disk is not mounted. The erasure of data for protection after loss is satisfied with user authentication function F2 and data protection function F3. Since all data are erased if an incorrect password is input more than 5 times, the leakage of data stored inside the flash drive is prevented even after the loss of a drive.

3.3. Security Evaluation. This section analyzes the security of the mechanism proposed in this paper. The analysis shows that the mechanism satisfies the confidentiality, integrity, authentication, and access control required by secure USB flash drives and safely protects the data from impersonation, man-in-the-middle, resending, and eavesdropping attacks by malicious attackers. The formalized verification tool AVISPA (Automated Validation of Internet Security Protocols and Applications) was used for the security analysis and the scenario based on the requirements and attack technologies described above.

3.3.1. Confidentiality. Secure USB flash drives require confidentiality in order to provide the internally stored data only to permitted users. Since the proposed mechanism uses the hash value of the user password to encrypt the whole file to be used as a disk, an attacker who does not have the password information cannot normally access the internally stored data.

3.3.2. Integrity. Secure USB flash drives must be able to guarantee integrity so that an unauthorized user cannot alter the internally stored data. Since the proposed mechanism uses the hash value of the user password to encrypt the whole file to be used as a disk, an attacker who does not have the password information cannot normally decrypt the data, and thus, integrity is assured.

3.3.3. Authentication. Secure USB flash drives must be able to authenticate the users to prevent unauthorized users from accessing the internally stored data and from normally accessing the internally stored data even if authentication is bypassed. The proposed mechanism authenticates users based on the ID and PW and encrypts the ID using the hash value of the PW. Therefore, the data are decrypted into unrecognizable data if an attacker inputs an arbitrary ID and PW to bypass the authentication. Since the disk is decrypted with the hash value of the arbitrary PW if an attacker bypasses authentication with the arbitrary data, the resulting disk data will be invalid, and thus, the internally stored original data will not be accessed.

3.3.4. Access Control. Secure USB flash drives must be able to prevent unauthorized users from accessing the internally stored data. Since the proposed mechanism encrypts the whole disk using the hash value of the user PW, an attacker who does not have the correct password or the hash value of the PW cannot obtain the correct decryption key and thus cannot decrypt the disk. Since the attacker cannot decrypt the disk, it is impossible to access the original data stored inside.

3.3.5. Impersonation Attack. Secure USB flash drives must be able to prevent an attacker from impersonating an authorized user so as to recover the internally stored data. Although an attacker may obtain the encrypted file header and the whole encrypted disk, an attacker who does not have the user PW cannot normally decrypt the disk since the decryption key is based on the user PW. Therefore, the proposed mechanism can protect the internally stored data from impersonation attacks.

3.3.6. Man-in-the-Middle Attack. Secure USB flash drives must be able to prevent man-in-the-middle attacks from extorting the authentication data and encryption/decryption key by inserting an additional module in the front part. Although an attacker may obtain the encrypted file header, encrypted ID, and encrypted disk file, the attacker will not have the information needed to decrypt the encrypted data since the user PW and the hash value of the PW used as the key are not transferred to the flash drive but are utilized only inside the management program. Therefore, the proposed mechanism is safe from man-in-the-middle attacks.

3.3.7. Eavesdropping Attack. Secure USB flash drives must be able to prevent eavesdropping attacks from obtaining and abusing the data transferred between the host and the flash drive. Since the proposed mechanism encrypts all data transferred between the host and the flash drive, it does not expose the authentication data and the encryption/decryption key and thus is safe from eavesdropping attacks.

3.3.8. Resending Attack. Secure USB flash drives must be able to prevent resending attacks which obtain the data transferred between the host and the flash drive and then resends them to bypass the authentication and access the data. Since the proposed mechanism only transfers the encrypted disk data from the flash drive, an attacker cannot bypass authentication and access the data even after obtaining the information. Therefore, the proposed mechanism is safe from resending attacks.

Lastly, the formalized verification tool AVISPA was used for the security analysis; the results of which showed that the proposed mechanism is safe. Figures 7, 8, 9, and 10 show the code used for the analysis and the analysis results.

```
protocol SecureUSB;
Identifiers
A, B                : user;
ID, FSH, DIF        : number;
Kps                 : symmetric_key; %Hashed PW

messages
1. A → B            : {ID, FSH}Kps
2. A → B            : {DIF}Kps
3. B → A            : {ID, FSH}Kps
4. B → A            : {DIF}Kps

knowledge
A                   : A, B, Kps:
B                   : A, B, Kps

session_instances
[A: program,B:usb,Kps:hashedpw];

goal
secrecy_of ID[];
secrecy_of DIF[];
A authenticates B on FSH;
```

Figure 7: AVISPA CAS code.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\hlpslGenFile.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS
  Analysed: 5 states
  Reachable: 3 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Figure 8: AVISPA ATSE result.

4. Conclusion

The USB flash drive is currently the most popular mobile storage unit because of its many strengths including fast data transfer speed, high portability, and free transfer and deletion. However, serious problems have arisen, such as the inability to protect the internally stored data after the loss of a USB drive, leading to demands for the development of a secure USB flash drive featuring improved security functions. For that reason, new and more secure USB flash drives protect the internally stored data using such security technologies as data encryption/decryption and user authentication

```
%OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\hlpslGenFile.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 10 nodes
  depth: 5 plies
```

Figure 9: AVISPA OFMC result.

```
SUMMARY
 INCONCLUSIVE
DETAILS
 NOT_SUPPORTED
PROTOCOL
 C:\progra~1\SPAN\testsuite\results\hlpslGenFile.if
GOAL
 SECRECY
BACKEND
 TA4SP
COMMENTS
 Some rules may be not fired so TA4SP does not do the verification.
STATICTICS
 Translation: 0.00 seconds
```

Figure 10: AVISPA TA4SP result.

and identification. However, the problems of access to the inside of a drive and the leakage of data have been identified in secure USB flash drives installed with the latest security technologies due to such vulnerabilities as implementation and environmental vulnerabilities, unlock command, and reverse engineering. To solve such problems, this paper proposes a safe secure USB flash drive mechanism that does not expose the authentication data. The mechanism overcomes the existing vulnerabilities to protect the data more safely, since it does not store the data needed for user authentication and disk decryption inside the flash drive data and has no routine for comparing the authentication. To analyze the security of the proposed mechanism, the security requirements which the secure USB flash drive must satisfy and an attack technology scenario were deduced. The results of the security assessment confirmed that the proposed mechanism satisfies the confidentiality, integrity, authentication, and access control requirements and safely protects the data from impersonation, man-in-the-middle, resending, and eavesdropping attacks. In addition, the formalized verification tool AVISPA was used for the security analysis; the results of which showed that the proposed mechanism is indeed safe.

Although the mechanism proposed in this paper was applied to a secure USB flash drive, it could be used in other areas. For example, it could be applied to a secure disk to improve the security of a hard disk and to secure backup storage so as to safely back up data and thereby protect internally stored data. Planned future studies include the application of a more improved mechanism, such as by adding device authentication to improve security through access control of the disk itself and by using the hash chain and a random number to supplement the weakness entailed by encryption with the same key.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

A part of this paper was presented at a conference on the International Symposium on Mobile Internet Security (MobiSec), October 19–22, 2017, Jeju Island, South Korea.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) that is funded by the Ministry of Education (NRF-2015R1D1A1A01057300) and by the Soonchunhyang University fund.

References

[1] Wikipedia, "USB Flash Drive," August 2016, https://ko.wikipedia.org/wiki/USB_%ED%94%8C%EB%9E%98%EC%8B%9C_%EB%93%9C%EB%9D%BC%EC%9D%B4%EB%B8%8C.

[2] K. Lee, K. Yim, and E. H. Spafford, "Reverse-safe authentication protocol for secure USB memories," *Security and Communication Networks*, vol. 5, no. 8, p. 845, 2012.

[3] O. Insu, Y. Lee, H. Lee, K. Lee, and K. Yim, "Study on secure USB mechanism without exposure of the authentication information," in *Proceedings of the International Symposium on Mobile Internet Security (MobiSec)*, Jeju Island, South Korea, October 2017.

[4] K. Lee, H. Yeuk, Y. Choi, S. Pho, I. You, and K. Yim, "Safe authentication protocol for secure USB memories," *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*, vol. 1, no. 1, pp. 46–55, 2010.

[5] J. Bang, B. Yoo, and S. Lee, "Secure USB bypassing tool," *Digital Investigation*, vol. 7, pp. S114–S120, 2010.

[6] S.-H. Lee, K.-B. Yim, and I.-Y. Lee, "A secure solution for USB flash drives using FAT file system structure," in *2010 13th International Conference on Network-Based Information Systems*, pp. 487–492, Takayama, Japan, September 2010.

[7] H. Jeong, Y. Choi, W. Jeon et al., "Vulnerability analysis of secure USB flash drives," in *2007 IEEE International Workshop on Memory Technology, Design and Testing*, pp. 61–64, Taipei, Taiwan, December 2007.

[8] J. Kim, Y. Lee, K. Lee, T. Jung, D. Volokhov, and K. Yim, "Vulnerability to flash controller for secure USB drives," *Journal of Internet Services and Information Security*, vol. 3, no. 3/4, pp. 136–145, 2013.

[9] K.-G. Lee, H.-W. Lee, C.-W. Park, J.-W. Bang, K.-y. Kim, and S. Lee, "USB PassOn: secure USB thumb drive forensic toolkit," in *2008 Second International Conference on Future Generation Communication and Networking*, pp. 279–282, Hainan Island, China, December 2008.

[10] S.-H. Lee, J. Kwak, and I.-Y. Lee, "The study on the security solutions of USB memory," in *Proceedings of the 4th International Conference on Ubiquitous Information Technologies & Applications*, pp. 1–4, Fukuoka, Japan, December 2009.

[11] A. N. Magdum and Y. M. Patil, "A secure data transfer algorithm for USB mass storage devices to protect documents," *International Journal of Emerging Engineering Research and Technology*, vol. 2, no. 4, pp. 113–119, 2014.

[12] L. Hamid, "Biometric technology: not a password replacement, but a complement," *Biometric Technology Today*, vol. 2015, no. 6, pp. 7–10, 2015.

[13] S.-H. Lee and I.-Y. Lee, "Secure index management scheme on cloud storage environment," *International Journal of Security and Its Applications*, vol. 6, no. 3, pp. 75–82, 2012.